PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-231053
(43)Date of publication of application : 05.09.1997

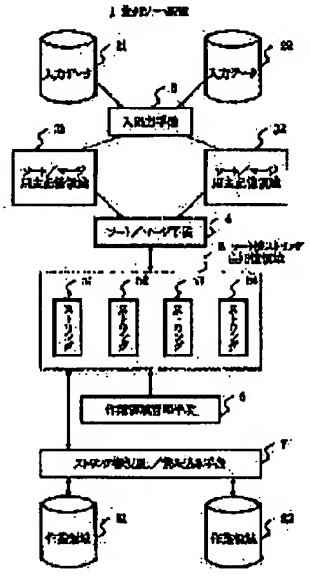
(51)Int.Cl. G06F 7/24
G06F 7/36
G06F 12/00

(21)Application number : 08-039799
(22)Date of filing : 27.02.1996

(71)Applicant : NEC SOFTWARE LTD
(72)Inventor : KITAZAWA ATSUSHI

(54) PARALLEL SORT DEVICE

(57)Abstract:
PROBLEM TO BE SOLVED: To shorten the parallel processing time of a parallel sort device by equalizing the numbers of sorted strings which are read out of a secondary storage in every parallel processing unit when the external sorts whose objects data are not completely stored in a main storage are carried out in parallel to each other in a multiprocess.
SOLUTION: The sort processing operations are carried out in the sort/merge main storage areas 31 and 32 which are proper to the parallel processes. Then these sort processing results are stored in a shared sorted string main storage area 5. Then the strings are written in the work areas 81 and 82, and a work area management means 6 decides a string replacement block when the area 5 has an overflow. The strings are taken out of the area 5 and sent to the areas 31 and 32 to perform the merging operations after the sort phase is over. Then a string writing/reading means 7 decides the blocks to be read next, and these blocks are read out of the areas 81 and 82.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平9－231053

(43)公開日　平成９年(1997)９月５日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ			技術表示箇所
Ｇ０６Ｆ	7/24			Ｇ０６Ｆ	7/24	Ｍ	
	7/36				7/36		
	12/00	５１２			12/00	５１２	

審査請求　有　　請求項の数２　ＯＬ　（全７頁）

(21)出願番号　　特願平8－39799

(22)出願日　　平成８年(1996)２月27日

(71)出願人　000232092
日本電気ソフトウェア株式会社
東京都江東区新木場一丁目18番６号

(72)発明者　北沢　敦
東京都江東区新木場一丁目18番６号　日本電気ソフトウェア株式会社内

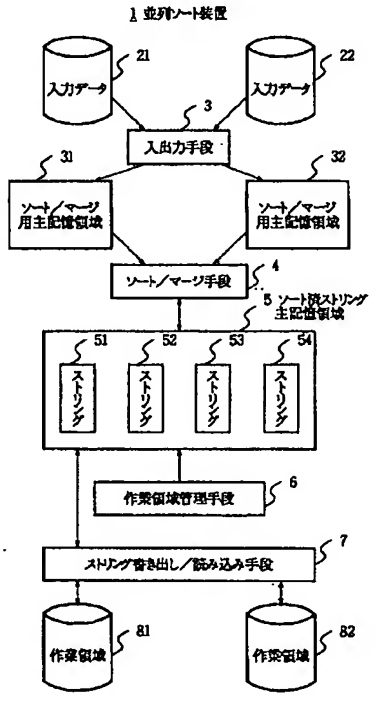
(74)代理人　弁理士　京本　直樹　（外２名）

(54)【発明の名称】　並列ソート装置

(57)【要約】

【課題】　対象データが全て主記憶に納まらない外部ソートをマルチプロセスで並列実行する際に、ソート済みストリングの二次記憶からの読み出し量を各並列処理単位で均等にすることで、並列処理時間を短縮する。

【解決手段】　並列処理プロセスに固有のソートマージ用主記憶領域３１，３２で、ソート処理を行い、結果を共有のソート済みストリング用主記憶領域５に格納する。作業領域８１，８２にストリングを書き出し、ソート済みストリング用主記憶領域５がオーバーフローした時、作業領域管理手段６がストリング置き換えブロックを決定し、ソートフェーズ終了後にソート済みストリング用主記憶５からソート／マージ用主記憶３１，３２にストリングを取り出しマージを行い、ストリング書き出し／読み込み手段７が次に読むべきブロックを決定し、作業領域８１，８２から読み込みを行う。



【特許請求の範囲】
【請求項１】　ソート対象の入力データを格納する第一の二次記憶と、
前記第一の二次記憶から読み出す入力データの所定数ブロックからなるストリングあるいはソート／マージ処理の中間段階で生成されるストリングを保持するソート／マージ用主記憶領域と、
ソート／マージ処理の中間段階で生成されるストリングの格納領域をオーバフローするブロックを退避させる作業領域を有する第二の二次記憶と、
を各プロセスそれぞれで具備し、
前記ソート／マージ用主記憶領域のストリングをソート／マージ処理して生成する新たなストリングを格納する前記格納領域からなるソート済ストリング主記憶領域と、
前記第一の二次記憶から入力データの所定数ブロックからなるストリングを読み出して前記ソート／マージ用主記憶領域に格納する入出力手段と、
前記ソート／マージ用主記憶領域のストリングをソート／マージ処理して生成した新たなストリングを前記ソート済ストリング主記憶領域に格納し、あるいは前記ソート済ストリング主記憶領域のストリングをソート／マージ処理して新たなストリングを前記ソート／マージ用主記憶領域に移送するソート／マージ手段と、を全プロセスで共有してなる並列ソート装置にあって、
前記ソート／マージ手段が前記ソート／マージ用主記憶領域のストリングをソート／マージ処理して生成する新たなストリングのブロックを前記ソート済ストリング主記憶領域のストリングに追加して生じるオーバフローブロックを前記作業領域に置き換えるとき、前記ストリングそれぞれの置き換えられるブロック数を等しくするように制御する作業領域管理手段と、
前記ソート済ストリング主記憶領域のオーバフローブロックを前記作業領域に前記置き換えのため書き出し、あるいは前記ソート済ストリング主記憶領域のストリングをソート／マージ処理するとき、前記置き換えによる空きブロックを、前記作業領域から読み出してストリングを復元するストリング書き出し／読み込み手段と、を備えることを特徴とする並列ソート装置。
【請求項２】　前記プロセスそれぞれをメモリ共有型マルチプロセッサの各プロセッサに割り当てることを特徴とする請求項１記載の並列ソート装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】この発明は、ソート／マージ処理の並列実行に関し、特に、ソート対象データが主記憶の共有領域と複数の二次記憶に分離格納されている場合のソーティングに関する。
【０００２】関係データベースシステム内部でソーティングが利用される場合、関係データを格納する二次記憶が分離格納されている場合や、導出された関係データを格納する、中間的に利用される主記憶と二次記憶が分離されている場合がある。
【０００３】近年実用化されている並列システムでは、主記憶を処理モジュールが共有するｓｈａｒｅｄ　ｅｖｅｒｙｔｈｉｎｇ型のアーキテクチャで、この発明は利用できる。さらに、主記憶を共有しないｓｈａｒｅｄ　ｎｏｔｈｉｎｇ型でも、分散共有メモリアーキテクチャを採用したシステムは、この発明を適用できる。
【０００４】
【従来の技術】この発明の関連分野の文献を挙げると、
［１］特開平０３－０７１２２７
［２］Ｂｉｔｔｏｎ　Ｄ．，ｅｔ　ａｌ：“Ｐａｒａｌｌｅｌ　Ａｌｇｏｒｉｔｈｍｓ　ｆｏｒ　ｔｈｅ　Ｅｘｅｃｕｔｉｏｎ　ｏｆ　ＲｅｌａｔｉｏｎａｌＤａｔａｂａｓｅ　ＡＣＭ　Ｔｒａｎｓａｃｔｉｏｎｓ　ｏｎ　Ｄａｔａ－ｂａｓｅ　Ｓｙｓｔｅｍｓ，Ｖｏｌ８，Ｎｏ３，Ｓｅｐｔｅｍｂｅｒ　１９８３，ｐａｇｅｓ３２４－３５３．
［３］樋川英治、渡部栄一：“ＮｏｎＳｔｏｐ　ＳＱＬのアーキテクチャ”，情報処理，Ｖｏｌ３３，Ｎｏ１２，ｐａｇｅｓ１４３６－１４４０がある。これらの文献を引用して、従来技術を説明する。
【０００５】文献［１］によれば、主記憶（容量ｍ）にソート対象となるデータ（容量ｎ）がすべて収まらない様な大容量のソート処理を外部的ソートと呼び、１回のソートフェーズとＬｏｇ（ｎ／ｍ）回のマージフェーズをもって構成される［文献１の４頁］。各フェーズでは、その中で完全にソートされた結果をもつ、いくつかのグループが生成する。このグループをストリングと呼ぶこととする。たとえば、ソートフェーズ完了時のストリングの個数は、最大ｎ／ｍである。外部ソートの性能は、ソートフェーズとマージフェーズあるいは各々のマージフェーズ間でデータ受け渡しをする作業領域に対するデータ転送時間が全体性能を制限する傾向にある（つまり、作業領域にたいするＩ／Ｏｂｏｕｎｄである）［文献１の８ページ］。［文献１］では、ソートフェーズとマージフェーズの受け渡しを、仮想記憶にマッピングすることで、仮想記憶をサポートするメモリ量が十分にある場合、３０％から８０％この時間を縮小する方法を開示している。この方法によれば、ソート処理に使用するメモリをソート／マージ処理用と作業領域支援用に等しく分割し、作業領域支援用のメモリは一つのストリングが処理された時点で（すなわちソート／マージ処理領域の全てのデータが処理された時点）、その都度、二次記憶に書き出し、次のマージフェーズでは主記憶に残っているストリングから処理することで作業領域からの読み出し時に発生する二次記憶へのＩ／Ｏ回数を削減するものである。なお二次記憶への書きだしは入力データの読み込みとオーバラップさせている。

【０００６】一方、外部ソートを複数プロセッサの並列マシンに適用した例が、文献［２］に示されている。図５は［文献２の頁３３４］から引用した並列２ウェイマージソートを説明する図で、ＳＵＢ　ＯＰＴＩＡＭＬ　ＳＴＡＧＥとＯＰＴＩＭＡＬＳＴＡＧＥが一つのデータ源に対する外部ソートに対応する。すなわち外部データ９１乃至９４をプロセッサ９５乃至９８でソートし、その結果を最終的に各プロセッサ９５乃至９８毎に１本のストリング９９にマージしている。並列２ウェイマージソートでは、個々の外部ソートの結果をさらに多段階に構成された複数プロセッサでマージすることで（ＰＯＳＴＯＰＴＩＭＡＬ　ＳＴＡＧＥ）、１本のソートされた結果を得ることができる。

【０００７】関係データベースシステムにおける関係代数演算処理を並列処理する場合、文献［３］によれば、関係を複数の二次記憶装置に分離して格納しておき、各々の分離した関係（パーティションと呼ぶ）に対して関係代数演算を施し、その結果をまとめる場合がある。この場合の一般的な考慮として、データの片寄りの問題がある。関係代数演算処理では、ソート処理の入力が通常選択処理等の別の関係代数演算の結果生じるので、データの片寄りは、単に各パーティションのデータ量が各々異なるだけでなく、パーティション毎の選択率が異なることによる片寄りもある。この様に並列処理する対象のデータ量に片寄りがある場合、並列ソートのＯＰＴＩＭＡＬ　ＳＴＡＧＥとＰＯＳＴＯＰＴＩＭＡＬ　ＳＴＡＧＥ間で最もデータ処理量の多いプロセッサの処理終了を待ち合わせる必要が発生することになり、全体性能を低下させる。

【０００８】

【発明が解決しようとする課題】メモリ共有型、あるいはクラスタ型のアーキテクチャを持つ並列マシン上で並列ソート処理を実行する場合の問題は、ＳＵＢＯＰＴＩＭＡＬ　ＳＴＡＧＥ，ＯＰＴＩＭＡＬ　ＳＴＡＧＥを個々の外部ソートとみなし、単純に文献［１］の方法を適用した場合、各外部ソートで発生する作業領域の二次記憶へのＩ／Ｏ回数が異なり、最も多くのＩ／Ｏを必要とする外部ソートの性能で、並列ソート全体の性能が抑制されることである。これはメモリ共有型のアーキテクチャの特徴を生かしていない。他の外部ソートのメモリを融通することで、Ｉ／Ｏ回数のバランスをとることが可能なはずだからである。

【０００９】外部ソートの場合、各フェーズで同一データに関する二次記憶へのＩ／Ｏ回数は高々２度である。１度は入力データの読み出し、あるいは前フェーズの作業領域からの読み出しであり、１度は作業領域へのデータの書き込みである。したがって、Ｉ／Ｏ回数を削減する方法は、フェーズ数を削減するか、あるいは各フェーズの作業領域へのデータの読み書きを主記憶でバックアップするかのいずれかである。フェーズ数を削減するためには、ソート／マージ用に割り当てる主記憶量を調整する必要がある。ソート／マージ用に割り当てられた主記憶はランダムにアクセスする傾向にあり、作業領域用に割り当てられた主記憶は順次アクセスとなる。

【００１０】この発明では、並列ソートのＳＵＢＯＰＴＩＭＡＬ　ＳＴＡＧＥ，ＯＰＴＩＭＡＬ　ＳＴＡＧＥを外部ソートとみなし、作業領域用に割り当てるメモリを並列ソートの間で融通しあうことで、マージフェーズのストリングで２次記憶から読み出すブロック数を各ストリングで均等にし、並列動作する外部ソートの終了時刻をなるべく同じにする。

【００１１】

【課題を解決するための手段】この発明の目的は、関係データベースの問合せ処理の並列化処理で、ソート処理を並列実行する際、二次記憶に対する入出力回数のバラツキで生じる処理時間の片寄りを均等に調整することによって、全体性能を向上させることにある。

【００１２】そのため、この発明の、ソート対象の入力データを格納する第一の二次記憶と、前記第一の二次記憶から読み出す入力データの所定数ブロックからなるストリングあるいはソート／マージ処理の中間段階で生成されるストリングを保持するソート／マージ用主記憶領域と、ソート／マージ処理の中間段階で生成されるストリングの格納領域をオーバフローするブロックを退避させる作業領域を有する第二の二次記憶と、を各プロセスそれぞれで具備し、前記ソート／マージ用主記憶領域のストリングをソート／マージ処理して生成する新たなストリングを格納する前記格納領域からなるソート済ストリング主記憶領域と、前記第一の二次記憶から入力データの所定数ブロックからなるストリングを読み出して前記ソート／マージ用主記憶領域に格納する入出力手段と、前記ソート／マージ用主記憶領域のストリングをソート／マージ処理して生成した新たなストリングを前記ソート済ストリング主記憶領域に格納し、あるいは前記ソート済ストリング主記憶領域のストリングをソート／マージ処理して新たなストリングを前記ソート／マージ用主記憶領域に移送するソート／マージ手段と、を全プロセスで共有してなる並列ソート装置にあって、前記ソート／マージ手段が前記ソート／マージ用主記憶領域のストリングをソート／マージ処理して生成する新たなストリングのブロックを前記ソート済ストリング主記憶領域のストリングに追加して生じるオーバフローブロックを前記作業領域に置き換えるとき、前記ストリングそれぞれの置き換えられるブロック数を等しくするように制御する作業領域管理手段と、前記ソート済ストリング主記憶領域のオーバフローブロックを前記作業領域に前記置き換えのため書き出し、あるいは前記ソート済ストリング主記憶領域のストリングをソート／マージ処理するとき、前記置き換えによる空きブロックを、前記作業領域から読み出してストリングを復元するストリング書き

出し／読み込み手段と、を備えることを特徴とする。

【００１３】ソートの対象となるデータは、二次記憶の入力データからソート／マージ用主記憶領域に入り切る所定数のブロックをストリングとして取り出され、その中で内部ソートされる。この処理は、別々のプロセス（あるいはプロセッサ）によって並列に実施される。いったん内部ソートが完了して生成される新たなストリングは個々のプロセスでソート済ストリング用主記憶領域に転送される。ソート済ストリング用主記憶のストリングのブロックは、二次記憶の作業領域に転送されて対応付けられる。ソート／マージ用主記憶領域に各々のプロセスで新たな入力データが読み出されるが、この際の待ち時間には、ストリング書きだし／読み出し手段がソート／マージ用主記憶領域のストリング中のブロックを書き出せるだけソート済ストリング主記憶領域と二次記憶の作業領域に書き出す。前記動作中に、転送すべきソート済ストリング用主記憶に空きがなくなった場合、作業領域管理手段が、各ストリングで置き換えブロック数が等しくなる様に置き換えるべきブロックを決定する。この決定方法については、実施例で説明する。

【００１４】マージフェーズでは、ソート済ストリング用主記憶領域の順次ソート済のストリングを順に取り出し、マージ用主記憶領域に転送してマージ処理を行う。これは、複数のプロセスで並列に実施される。この動作を各プロセスが作成した全てのストリングに対して実施する。マージ処理が完了したものは、従来技術で説明したＰＯＳＴＯＰＴＩＭＡＬ　ＳＴＡＧＥの処理に渡される。これは、同一マシン内で実施してもよいし、別マシンに転送して実施してもよい。

【００１５】上記動作中に、ソート済ストリング用主記憶のブロックに空きができた場合、ストリング書き出し／読み込み手段は二次記憶の作業領域から空きになったブロックと同じストリングで次に利用するブロックの読み出し要求を出す。どのブロックを読み出すかについての方法を実施例で説明する。二次記憶の作業領域からブロックの読み出しは、マージ処理実行中に行われる。マージ処理は、ソート済ストリング主記憶領域中のストリングがなくなった場合、ストリング書き出し／読み出し手段が要求した読み出し要求の終了を待ち合わせる。この動作によって、各プロセスが二次記憶の作業領域から読み出すブロック数を均等にして、プロセス間で外部ソートに必要な処理時間を合わせることができる。

【００１６】

【発明の実施の形態】次に、この発明について、図面を参照して説明する。

【００１７】この発明の一実施例の構成を示す図１を参照すると、並列ソート装置１が２つのプロセスで動作する構成を例示し、ソートの入力となる入力データ２１，２２と、入力データをプロセス対応のソートマージ用主記憶領域３１，３２に読み込むための入出力手段３と、読み込んだ入力データをソート／マージ用主記憶領域３１，３２上でソートあるいはマージして新しいストリングを生成するソート／マージ手段４と、ソート／マージ手段４によってソートされた結果であるストリング５１乃至５４を蓄えるソート済ストリング用主記憶領域５と、入力データの量が多く、ソート済ストリング領域５がオーバフローして、該主記憶領域５が不足した場合に備えて、ストリング５１乃至５４でどのブロックを新たなブロックの置き換え用として選択するかを決定する作業領域管理手段６と、作業領域管理手段６が決定したブロックをプロセス対応の作業領域８１，８２に書き出すストリング書き出し／読み込み手段７と、を備える。

【００１８】この実施例では、各並列処理のプロセスがソート結果を１本のストリングとしている。これは、従来技術で述べた並列ソートのＳＵＢＯＰＴＩＭＡＬ　ＳＴＡＧＥ，ＯＰＴＩＭＡＬ　ＳＴＡＧＥに相当する。最終的に各プロセスで処理したソート済ストリングから１本のソート結果を得るには、さらにその結果をマージする必要がある。入力データ２１，２２とソート／マージ用主記憶領域３１，３２と、作業領域８１，８２は並列処理単位のプロセス毎に存在し、ソート済ストリング用主記憶領域５のみが、並列処理のプロセスに共通した領域であるものとしている。作業領域８１，８２や、ソート／マージ用主記憶領域３１，３２を並列処理単位に共通にする変更を加えることは可能である。

【００１９】次に、この実施例のソートマージ用主記憶領域３１，３２と、ソート済ストリング用主記憶領域５のサイズの決定方法を説明する。更に作業領域管理手段６がソート済ストリング用主記憶領域５上のストリング５１乃至５４のどのブロックを置き換えるかの決定方法を説明し、更にソート済ストリング用主記憶領域５に空きブロックが発生した場合、ストリング書き出し／読み出し手段７がどのブロックを二次記憶の作業領域８１，８２から復元するかの決定方法を説明する。

【００２０】先ず主記憶領域のサイズの決定方法を説明する。ソート／マージ用主記憶はプロセス固有の領域であり、サイズは各プロセスで同じにする。各プロセスが処理する入力データサイズをｎとし、ソート用主記憶のサイズをｍとすると、一つのプロセスが生成するソート済ストリングの数は最大でｎ／ｍとなる。一方、マージ処理が必要とするメモリ量は、処理対象のストリング数に依存して増加する。簡単のために、マージ対象のストリングは、そのプロセスが生成したストリングであるとする。マージ処理で必要な主記憶量をｆ（ｎ／ｍ）で表わすと、

ｍ＝ｆ（ｎ／ｍ）

なるメモリ量ｍを決定することが望ましい。たとえば、ｆ（ｎ／ｍ）＝ｎ／ｍのケースを考えると、上式はｍ＝ｎ／ｍとなり、ｍ＝ｒｏｏｔ（ｎ）となる。上述のように、ソート／マージ用主記憶領域サイズは、マージアル

ゴリズムで使用するメモリ量と処理の入力データとから決定することができる。入力データのデータ量の半分の桁数のメモリ量を割り当てることになる。利用可能な主記憶のうち、残りの主記憶をソート済ストリング用主記憶領域に割り当てる。

【００２１】次に、作業領域管理手段６のブロック置き換えの動作の流れ図３を参照し、ブロック置き換えの動作を説明する図２（ａ），図２（ｂ），図２（ｃ），図２（ｄ）を援用して、作業領域管理手段６による置き換えアルゴリズムを説明する。図２（ａ）乃至図２（ｄ）では、ソート処理結果としてストリング５１乃至５４の４本が生成され、ソート済ストリング用主記憶領域５に渡されるものとしている。ソート済ストリング用主記憶領域５に存在するブロックは全部で１２ブロック分であるものとしている。また１２ブロックがオーバフローして、ブロックの置き換えが発生し、ソート済ストリング用主記憶領域５から作業領域８１，８２に置き換えで追い出されたブロックは黒の塗り潰しで示すことにする。図２（ｃ），図２（ｄ）で、置き換え数とは、この黒塗りのブロック数である。また、残りとは、図２（ａ），（ｂ）で網点で示されているソート済ストリング主記憶領域５の１２ブロック内に残っている数である。空白のブロックは、今後処理で生成される予定のストリングのブロックである。ストリング５１とストリング５２はプロセス１５がソート処理したストリングであり、後のマージ処理でもプロセス１５が使用する。ストリング５３とストリング５４はプロセス１６がソート処理したストリングであり、後のマージ処理ではプロセス１６が使用する。

【００２２】先ず図２（ａ）は、ストリング５３の生成時に領域がオーバフローし、初めて置き換えが必要になった時点でソート済ストリング主記憶領域５の１２ブロック内にストリングが納められた状況を示している。この時点で、ストリング５１，５２，５３の全ての置き換え数は０であるが、生成したブロックを該領域５に格納するとき、図３によれば、自分以外（つまり処理対象となっているストリング５３以外）を優先的に選択するので、置き換え対象としてストリング５１が選ばれる（ステップ６１）。ストリング５１の１ブロックを置き換えた場合にソート済ストリング用主記憶に残るブロックが存在する（１以上である）ので、このストリングの最後に転送された部分（図２の５１３）が置き換えの対象となったことを示している（ステップ６２のｎｏ）。なお、ブロック１個を残すのは、全てのソートストリングが生成された後のストリングのマージ処理を実行するためには、全てのストリングの最も小さい値を含むブロックがソート済ストリング用主記憶領域５に存在する必要があるからである。図３のアルゴリズムにしたがって、４本全てのストリングの処理が終わった時点の状態を図２（ｂ）に示す。ストリング５４のブロックは、自身のストリング書き出し中に発生した黒の塗潰し部分５１２，５１３，５２２，５２３，５３６，５３７，５４２，５４６の置き換えによって、１２ブロック内の網点５１１，５２１，５３１乃至５３５，５４１，５４３乃至５４５，５４７に納められている。この様な置き換えは、図３のステップ６２の条件から、図２のブロック５４１では発生しない。図２（ｂ）によれば最終的に全てのストリングで置き換え数が２となり、読み出すべきブロック数が同じになる様に調整されていることがわかる。このことは、マージ処理時にプロセス１５およびプロセス１６で必要な作業領域８１，８２からの読みだし回数が等しくなり、各プロセスでのマージ処理終了時間が均等になることを意味する。ソート済ストリング主記憶領域５にストリングが生成されて、ブロックを置き換えるためには、ソート済ストリング用主記憶５の置き換え対象となるブロックをストリング書き出し／読みだし手段７で作業領域８１，８２に書き出す必要がある。書き出すタイミングを、置き換えが発生するタイミングとしたが、事前に書き出しておくこともできる。たとえば、最初に置き換えが発生する一つ前（つまり、ソート済ストリング用主記憶領域に１ブロックの空きがある時点で）に、図３のアルゴリズムを適用し、次に置き換えが発生した場合に置き換えるブロックの書き出し要求を出しておくことで、入力データからの読み出しおよびソート処理と作業領域８１，８２へのブロックの書き出し処理をオーバラップさせて、処理時間を短縮することができる。マージフェーズでソート済ストリング領域５の或るブロックの処理が完了し、そのブロックが再利用可能となった場合、ストリング書き出し／読み込み手段７がどのブロックを作業領域８１，８２から読み出すかを説明する。

【００２３】先ず図２（ｂ）がマージ処理の開始時点である。ストリング書き出し／読み込み手段７のブロック選択基準を示すフローチャートである図４によれば図２（ｂ）で、プロセス１６のマージ処理が進行し、ブロック５３１が空になったとする。ここで、図４の１のフローチャートに従うと、プロセス５６自身のマージ処理で、空きのブロック５３１に読み出す作業領域のブロックは、図２（ｂ）の５３６，５４２から置き換えられたブロックで、このうち５４２は距離１であり、５３６は距離５であることから（ステップ７１，７２）、ストリング書き出し／読み込み手段７はブロック５４２を読み出す（ステップ７３）。この間、マージ処理は、図２（ｂ）のブロック５４１，５３２の間で継続する。一方、プロセス１５のマージ処理が進行しブロック５１１が空になったとすると、次に読み出すべきブロックは図２（ｂ）の５１２，５２２で置き換えられたブロックであり、各々距離は１なので、自分のストリングの次のブロックである５１２が読み出しの対象となる。

【００２４】なお、この実施例では、ダブルバッファに

よる先読みを考慮していないが、図3のステップ2で、残すバッファ数を調整することで、ダブルバッファによる先読みを実施することもできる。たとえば、図3のステップ2の判定条件を、「残りが1以下または、そのプロセスの先読み用バッファが1以下」あるいは、「残りが2以下」などと変更すればよい。この場合、図4のアルゴリズムはまったく変わりない。またこの実施例において、プロセスを共有メモリ型マルチプロセッサの各プロセッサに割り当てる変更ができることは明らかである。

【0025】

【発明の効果】以上説明した様に、この発明によれば、並列に実行する外部ソートのマージフェーズで読み出すべきブロック数を均等になるように調整することによって、マージフェーズの終了時刻を各プロセスで同じにして、ソート／マージ処理全体の処理時間を短縮できる。たとえば、図2（b）の例では、プロセス毎に別々の作業領域用の二次記憶を持つとすると、二次記憶へのI／O回数は各プロセスで4回ずつである。これに対して、各プロセスで均等にソート済ストリング用主記憶を割り当てたとすると、プロセス15は全てメモリに収まるために、I／O回数は0であり、プロセス16では8回のI／Oが必要となり、この発明の装置の2倍のI／O回数のプロセスが全体の処理時間を低下させる。

【0026】また、この発明によれば、空きブロックに読み出すブロックを決定する際に、現在使用しているブロックからの距離を利用することで、最も必要度の高いデータを読み出しの対象とすることができ、I／O中にマージ処理を並列して行うことができる。たとえば、図2（b）の例では、ブロック531が空となった場合にブロック536を読み出すと、次にブロック542が必要になった場合に、ブロック542の読み出し完了までマージ処理を待つ必要がある。これは、ブロック542がブロック536よりも前に必要となる可能性が高いから各ストリングが均等なソートキーの分布を持つ場合には有効になる。

【図面の簡単な説明】

【図1】この発明の一実施例の構成を示す図である。

【図2】分図（a）は、ストリングに新たなブロックをつなぐとき、置き換えるブロックを決定する説明図、分図（b）は、ソート済ストリングをマージするとき、読み出すブロックを決定する説明図、分図（c）は、分図（a）の時点でのストリングの状況を説明する図、分図（d）は分図（b）の時点でのストリングの状況を説明する図である。

【図3】ソート済ストリングを主記憶領域で置き換える際のフローチャートである。

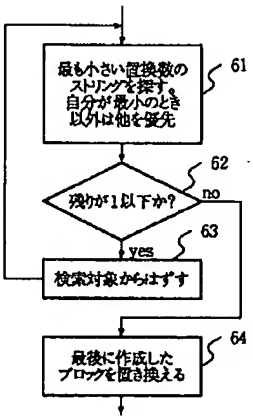
【図4】ソート済ストリング用主記憶領域に空ができた場合の二次記憶の作業領域からの読み出しのフローチャートである。

【図5】従来の並列ソートを示す図である。

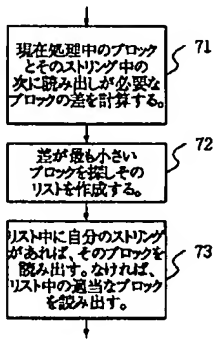
【符号の説明】

1　並列ソート装置
3　入出力手段
4　ソート／マージ手段
5　ソート済ストリング主記憶領域
6　作業領域管理手段
7　ストリング書き出し／読み込み手段
21，22　入力データ
31，32　ソート／マージ用主記憶領域
81，82　作業領域

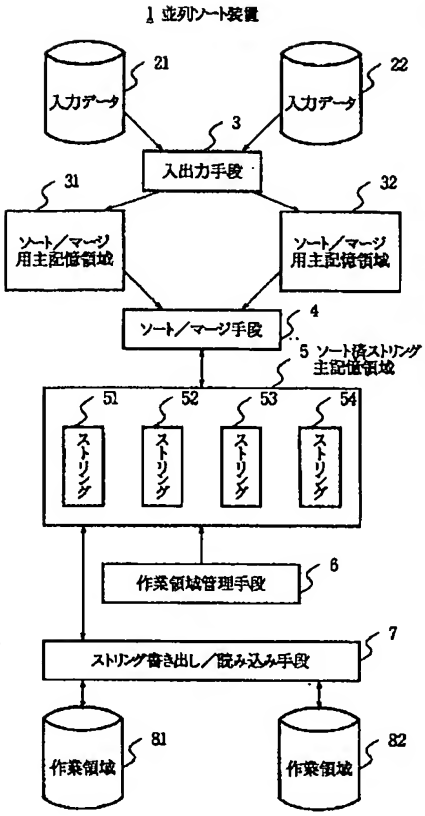
【図3】



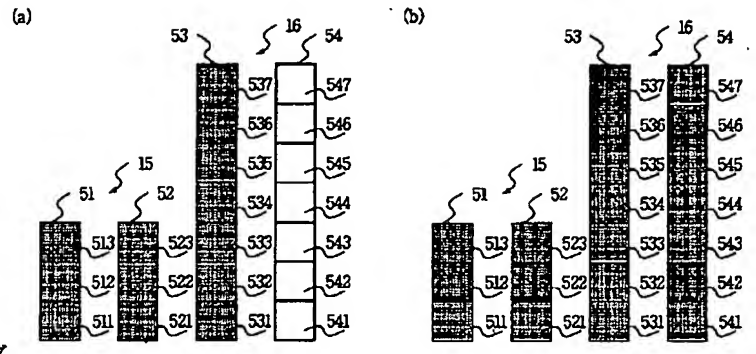
【図4】



【図1】



【図2】



(c)

	ストリング51	ストリング52	ストリング53	ストリング54
置換数	1	0	0	0
残り	2	3	7	0

(d)

	ストリング51	ストリング52	ストリング53	ストリング54
置換数	2	2	2	2
残り	1	1	5	5

15, 16 プロセス　　51, 52, 53, 54 ストリング　　511乃至513, 521乃至523, 531乃至537, 541乃至547 ブロック

【図5】